

nap

Nascom Assembler Package

Copyright (C) Juli 1980 Anders Hejlsberg Software



INDHOLDSFORTEGNELSE:

Introduktion	1
Intruduktion til manualen	2
Sammenligning med ZILOG assembleren	3
Hvad kræves af dit system	4
NAP betjening	4
NAP kommandoer	5
Editoren	6
Editorkommandoer	7
Assembleren	10
Talværdier	11
Symboler	12
Assemblerudskrift	13
I tilfælde af fejl	14
APP. A Indlæsning og opstart	15
APP. B Fejlnumre	16
APP. C Pseudoinstruktioner	18
APP. D Interne registre	19
APP. E Brugerdefinerbar udskriftsrutine	19
APP. F Et programeksempel	20
APP. G Kommandooversigt	22

INTRODUKTION:

NAP (Nascom Assembler Package) er en kombination af en assemblercompiler og en editor, specielt designet til NASCOM 1/2 microcomputerne.

Assembleren kan oversætte Z80 mnemonics til maskinkode, der både kan udlæses på bånd og direkte i lageret.

Editoren benyttes til indlæsning og editering af kildeteksten. Trods programmets ringe størrelse (4k bytes), har denne skærmorienterede editor mange funktioner, der normalt kun findes i langt større programmer, fx.:

- * BLOK KOMMANDOER, der giver mulighed for at kopiere, flytte eller slette større dele af kildeteksten.
- * TABULERINGSFUNKTIONER. Ved hjælp af disse kan programmet stilles op i kolonner allerede under indlæsningen, hvilket gør det langt mere overskueligt.
- * BLANKTEGNSKOMPRIMERING, der gør at et vilkårligt antal blanktegn efter hinanden kun fylder een byte i lageret.

INTRODUKTION TIL MANUALEN:

Manualen benytter følgende notationer og forkortelser:

- * Tal der ikke er i decimalnotation (10-talsnotation) efterfølges af et bogstav:

- H -- efterfølger et hexadecimalt tal
 - Q -- efterfølger et oktal
 - B -- efterfølger et binært tal

- * CTRL/ eller SHIFT/ efterfulgt af et bogstav, betyder at karakteren indtastes samtidig med at control- eller shiftnøglen holdes nede.

- * Specielle nøgler på NASCOM 2 tastaturet:

- VP -- Venstrepil (CTRL/Q)
 - HP -- Højrepil (CTRL/R)
 - OP -- Opad pil (CTRL/S)
 - NP -- Nedad pil (CTRL/T)

Manualen giver ingen generel instruktion i assembler-programmering, da det ville være alt for omfattende.

I stedet kan der henvises til følgende bøger om emnet:

Lance A. Leventhal
Z80 Assembly Language Programming
(Osborne & Associates, Inc.)

Adam Osborne et al
Z80 Programming for logic design
(Osborne & Associates, Inc.)

Rodnay Zaks
Programming the Z80
(Sybex)

MOSTEK
Z80 Programming Manual

Nat Wadsworth
Z80 Instruction Handbook
(Scelbi)

SAMMENLIGNING MED ZILOG ASSEMBLEREN:

På de fleste punkter er ZILOG assembler og NAP identiske.
Følgende bør dog iagttages:

- * Udtryk må arbejdes med følgende operatorer:

+ Addition	- Subtraktion
* Multiplikation	/ Division
& Logisk OG (AND)	. Logisk ELLER (OR)

Paranteser må kun benyttes til at indikere adresser, og kan således ikke indgå i udtryk.
Et udtryk evalueres fra venstre mod højre, uanset hvilke operatorer der benyttes.

- * Hexadecimale tal skal efterfølges af et 'H', oktal af et 'Q' og binære tal af et 'B'. Hvis ingen af disse bogstaver er tilstede betragtes tallet som decimalt.
- * Symboler og labels skal efterfølges af et kolon (':') når de defineres.
- * Følgende pseudoinstruktioner må benyttes:

DB	DW	DS
EQU	ORG	END
MEM		

Disse er nærmere beskrevet i APPENDIX C

- * To specielle instruktioner er blevet implementeret. Disse er:

RCAL label Denne instruktion genererer en RST 10H maskinkode, efterfulgt af een byte der angiver den relative adresse på den subroutine der skal kaldes.

SCAL kode Denne instruktion genererer en RST 18H maskinkode, efterfulgt af nummeret på den subroutine i NAS-SYS der skal kaldes.

Disse instruktioner er ikke standard i Z80 processoren og kræver derfor at systemet benytter NAS-SYS, eller en anden monitor med tilsvarende funktioner.

HVAD KRÆVES AF DIT SYSTEM:

- * Minimum 8k bytes lager
- * NAS-SYS Monitor
- * Kassettebåndoptager
- * TV eller monitor

NAP BETJENING:

Når NAP er indlæst og opstartet som beskrevet i APPENDIX A, udskrives der et lighedstegn (=) for at indikere at programmet venter på en kommando.

Alle kommandoer består, i lighed med NAS-SYS, af et enkelt bogstav (en undtagelse er dog 'S' kommandoen).

Kommandoindtastninger afsluttes med et tryk på ENTER. Fejlindtastninger kan rettes med BS nøglen.

Ved et tryk på ENTER alene, slettes skærmen.

NAP KOMMANDOER:

H HOWBIG, udskriver kildetekstens start- og slutadresse, og symboltabellens slutadresse.

Q QUIT, returnerer til NAS-SYS. Man kan quit-te og varm-starte (ved at skrive E1002) lige så ofte man vil, uden at det berører kildetexten eller NAP.

R READ, indlæser en kildetekst fra bånd. Den indlæste tekst bliver lagt i lageret efter den nuværende. Dette giver mulighed for at indlæse og sammensætte flere "moduler" til en større kildetekst.

Når båndteksten er fundet, bliver dens overskrift udskrevet på skærmen. Under indlæsningen udskrives hver enkelt linie og forsvinder igen efter et kort øjeblik, medmindre der optræder en checksumfejl. Sker dette, vil den forkerte linie blive scrollet op, og den kan senere rettes ved hjælp af 'E'-kommandoen.

S SORT, udskriver den symboltabel der blev opbygget ved sidste assemblering. Tabellen sorteres alfabetisk efter første bogstav. Efter kommandokaldet skriver NAP:

Option: _

Her kan der vælges mellem to muligheder:

- V Udskrift på skærmen.
- E Udskrift til brugerdefineret udskriftsrutine.

En udskrift kunne eksempelvis se således ud:

CRT	0065	KBD	0061	MRET	005B
NIM	0072	SRLIN	0070	START	0E00
STOP	0E2D	TEST	0E0B		

Af udskriften ses fx. at symbolet 'MRET' har værdien 005BH.

Der er også mulighed for at søge på et enkelt bogstav. Således vil kommandoen 'SS' kun udskrive de symboler der starter med 'S':

SRLIN	0070	START	0E00	STOP	0E2D
-------	------	-------	------	------	------

Symbolernes værdi bliver altid udskrevet i hexadecimal notation.

Ved kald af 'A', 'E', 'R' og 'Z' kommandoerne slettes symboltabellen.

W WRITE, udlæser kildetexten til båndoptageren. Efter kommandokaldet spørger NAP:

Name: _

Her kan der indtastes en overskrift (Max. 40 karakterer), der vil blive udskrevet når man indlæser båndet igen.

Z ZAP, sletter hele kildeteksten.

EDITOREN:

Editoren kaldes ved hjælp af 'E' kommandoen i NAP.

Som tidligere nævnt er editoren skærmorienteret. Dette betyder at skærmen fungerer som et 'vindue', der kan flyttes frem og tilbage over kildeteksten.

Cursoren befinder sig altid inden for dette vindue og angiver ved sin position, hvor der skal indsættes, rettes eller fjernes en linie eller nogle karakterer.

Denne opbygning har mange fordele frem for andre editortyper, blandt andet er der intet behov for linienumre, hvilket sparer mellem 10 og 20 procent lagerplads.

Før den editerede tekst bliver gemt i lageret bliver alle blanktegn komprimeret. Dette giver mulighed for at opstille teksten i kolonner under indtastningen, idet et vilkårligt antal blanktegn efter hinanden kun fylder een byte i komprimeret tilstand.

EDITOR KOMMANDOER:

CTRL/B	BEGIN, udskriver de første 15 linier af kilde- teksten. Cursoren flytter til HOME positionen, dvs første linie, første position.
CTRL/E	END, flytter cursoren til sidste linie af kilde- teksten.
CTRL/N	NEXT PAGE, flytter cursoren 14 linier frem i kildeteksten.
CTRL/O	OLD PAGE, flytter cursoren 14 linier tilbage i kildeteksten.
CTRL/F	FIND, finder den første forekomst af en søgetekst på max. 40 karakterer. Idet CTRL/F nedtrykkes indsætter NAP en linie på skærmen og udskriver en højrepil. Herefter kan søgeteksten indtastes (eventuelt under brug af BS-nøglen) og afsluttes ved et tryk på ENTER, hvorved linien forsvinder igen. Hvis teksten bliver fundet under søgningen, bliver den linie den står i placeret øverst på skærmen og cursoren går til HOME. Hvis teksten ikke bliver fundet, bliver cursoren stående. Søgningen starter altid i linien efter den nuværende.
CTRL/C	CONTINUE, fortsætter søgningen efter den sidst indtastede søgetekst.
OP CTRL/S	Flytter cursoren een linie tilbage i teksten. Hvis cursoren står i øverste linie scroller skærmen een linie ned.
NP CTRL/T	Flytter cursoren een linie frem i teksten. Hvis cursoren står i nederste linie scroller skærmen een linie op.
SHFT/OP CTRL/Y	Sletter den linie cursoren står på, og scroller de understående linier een linie op.
SHFT/NP CTRL/Z	Scroller den linie cursoren står på og de under- stående linier een linie ned, og indsætter en tom linie.
ENTER	Flytter cursoren een linie ned og indsætter en tom linie.
ESC	Sletter den linie cursoren står på, og flytter cursoren een linie op.

VP CTRL/Q	Flytter cursoren een position til venstre. Hvis cursoren står i første position ignoreres VP.
HP CTRL/R	Flytter cursoren een position til højre. Hvis cursoren står i sidste position ignoreres HP.
SHFT/VP CTRL/U	Fjerner den karakter cursoren står oven i, og flytter resten af linien een position til venstre.
SHFT/HP CTRL/V	Flytter den del af linien der står efter cursoren een position til højre, og indsætter et blank-tegn.
BS	Flytter cursoren een position tilbage og udskriver et blanktegn. Hvis cursoren står i første position ignoreres BS.
CH CTRL/W	Flytter cursoren frem til næste tab-position. Disse er som følger: Pos. 9 -- Start af instruktionsfeltet Pos. 14 -- Start af operandfeltet Pos. 26 -- Start af kommentarfeltet Pos. 47 -- Sidste position på linien Hvis man fx. står i position 1-8 vil et tryk på CH flytte cursoren til position 9.
CS CTRL/L	Sletter alle karakterer i det TAB-felt man befinder sig i, og flytter cursoren til begyndelsen af dette.
LF CTRL/J	Flytter cursoren til første position på linien.
CTRL/;	START BLOK, rykker hele den linie cursoren står i een position til højre, og indsætter et startblok mærke i første position.
CTRL/=	SLUT BLOK, rykker hele den linie cursoren står i een position til højre, og indsætter et slutblok mærke i første position.
CTRL/D	DELETE BLOK, fjerner den første mærkede blok (og blokmærkerne) fra kildeteksten. Cursoren flytter til første position i linien hvor slutblok mærket stod, og placerer denne i første linie på skærmen.
CTRL/I	INSERT BLOK, indsætter den første mærkede blok (men ikke blokmærkerne) før den linie cursoren står i. Cursoren flytter sig ikke. Hvis cursoren står i en af linierne mellem blokmærkerne, eller i den linie hvor slutblok mærket står, ignoreres CTRL/I.
CTRL/P	PRINT BLOK, udskriver den første mærkede blok via den brugerdefinerbare udskriftsrutine.

CTRL/X

Sletter skærmen og går tilbage til kommandoniveauet. Når man igen kalder editoren, kommer man tilbage til det samme sted i kildeteksten, medmindre 'A'-kommandoen har været brugt.

For at lette editeringen har NAP et indbygget underprogram, der giver repeterende tastatur.

I modsætning til nogle systemer, der bruger en speciel repetitionstast, repeterer tastatuets automatisk den nedtrykkede tast efter en kort pause. Dette program er listet som program eksempel i APPENDIX F.

ASSEMBLEREN:

NAP assembleren kan oversætte Z80 instruktioner som de er defineret i ZILOG Z80 Assembly Language Programming Manual.

Hver linie kildetekst kan som regel opdeles i 4 felter:

LABELFELT, INSTRUKTIONSFELT, OPERANDFELT, KOMMENTARFELT

Som eksempelvis denne linie:

START: LD HL,(SLUT) ;Indlæs slutadresse

Felterne skal adskilles af mindst et blanktegn.

LABELFELTET

bruges kun når man definerer et symbol og er ellers tomt.

INSTRUKTIONSFELTET

bruges til instruktionen. Der er 74 instruktioner og derudover 7 pseudoinstruktioner, som bliver forklaret senere.

OPERANDFELTET

Inholdet af dette felt afhænger af instruktionen. En instruktion som JR har enten en eller to operander (JR LABEL eller JR Z,LABEL), mens instruktionen NOP ingen operander har. I det sidste tilfælde er dette felt altså tomt.

KOMMENTARFELTET

strækker sig over resten af linien, men skal ikke nødvendigvis bruges. Ønsker man at skrive en kommentar skal den altid stå efter et semikolon, hvorved assembleren ignorerer resten af linien.

Denne inddeling har dog visse undtagelser, idet kommentarer godt kan strække sig over en hel linie. Ligeledes kan helt tomme linier også forekomme, hvilket ofte bruges til at lave afsnit i kildeteksten.

TALVÆRDIER:

Talværdier bruges ofte som operander, fx.

```
LD  A,10H
```

Følgende bliver evalueret som talværdier:

TAL

Assembleren forstår tal i hexadecimal, decimal, oktal og binær notation.

Tal skal altid begynde med et ciffer (0-9).

BOGSTAVER

Assembleren kan generere den binære værdi af et hvilket-somhelst ASCII-tegn, blot det er indesluttet i enkelte eller dobbelte anførelsestegn.

SYMBOLER

Disse er beskrevet i et senere afsnit.

PROGRAMPILEN

er en intern tæller der simulerer microprocessorens program counter (PC). Programpilen har altid den værdi der svarer til instruktionens adresse, og kan benyttes ved hjælp af et dollartegn ('\$').

UDTRYK

De ovennævnte datatyper kan ved hjælp af forskellige operatorer sammensættes til udtryk. Disse operatorer er:

- + Addition
- Subtraktion
- * Multiplikation
- / Division
- & Logisk OG (AND)
- . Logisk ELLER (OR)

Eksempler på dette er:

```
LD  HL,VRAM+14*64
LD  A,'Q'.11000000B
CP  ADDR&255
JR  Z,$+10
```

Udtryk bliver udregnet fra venstre mod højre uanset hvilke operatorer der bruges.

SYMBOLER:

Symboler er en af de grundlæggende elementer i assembler-sproget, og består af et navn der er tilskrevet en værdi. Denne værdi kan bestemmes på to måder:

- 1) Ved hjælp af EQU-pseudoinstruktionen. Fx.

```
VRAM: EQU 080AH
```

Denne linie giver symbolet VRAM værdien 080AH, og hvergang der bliver refereret til VRAM under assembleringen vil denne værdi blive indsat.

- 2) Ved at placere symbolet før en instruktion. Assembleren vil da give symbolet en værdi der svarer til adressen på instruktionen.

Ligegyldigt hvilken metode der benyttes, skal et symbol altid efterfølges af et kolon, når det defineres.

Et symbol skal begynde med et bogstav, man kan derefter indeholde både bogstaver og tal. Teoretisk set kan symboler have hvilken som helst længde, men normalt bør den ikke overstige 6 karakterer, da dette ødelægger symmetrien når tab-funktioner benyttes.

Reserverede ord, så som instruktioner, registernavne og condition codes, må ikke benyttes som symboler, men må gerne indgå i symbolets navn.

ASSEMBLERUDSKRIFT:

Når assembleren kaldes, spørges der:

Option: _

Her kan der vælges mellem 4 muligheder:

- V VIDEO, giver assemblerlistning på skærmen.
- E EXTERNAL, giver assemblerlistning til den brugerdefinerbare udskriftsrutine (se APPENDIX E).
- C CASSETTE, giver ingen assemblerlistning, men dumper i stedet objekt-koden til den kassettebånd-optageren. Denne udlæsning kan senere indlæses med NAS-SYS 'R'-kommando.
- ENTER Giver hverken assemblerlistning eller objekt-kode-dump. Denne metode er den hurtigste, og bør benyttes indtil kildeteksten er fejlfri.

Når der listes til skærmen ('V'), er sidelængden 14 linier. Efter hver side udskriver NAP et punktum på skærmen og blinker med cursoren, indtil en tast nedtrykkes.

I TILFÆLDE AF FEJL:

Hvis assembleren finder en fejl i kildeteksten sker følgende:

- Assembleringen stopper
- Der udskrives en fejlmeddelelse
- Den linie der er fejl i bliver den linie cursoren står ved næste gang editoren kaldes
- Linien udskrives
- NAP afventer en kommando

Kald da editoren, ret fejlen, og assembler igen.

APPENDIX A -- INDLÆSNING OG OPSTART

NAP leveres på et kassettebånd der er udlæst med NAS-SYS 'W' kommando på 300 baud. Hvis det første indlæsningsforsøg fejler er der en udlæsning mere på den anden side af båndet.

NAP koldstartes ved at indtaste:

E1000 bbbb eeee (ED000 bbbb eeee for ROM versionen)

hvor bbbb er tekstbufferens startadresse og eeee er den sidste adresse der må benyttes af NAP. Hvis eeee udelades benyttes alt det lager der er tilgængeligt over adressen bbbb. Hvis bbbb udelades starter tekstbufferen i 2000H (1000H for ROM versionen).

NAP varmstartes i 1002H (D002H for ROM versionen).

APPENDIX B -- FEJLNUMRE

ERROR 00 SYNTAKS FEJL

Symbol, instruktion eller operand indeholder ulovlige karakterer, eller et semikolon (;) mangler før en kommentar.

ERROR 01 RESERVERET ORD

Symbolet er et reserveret ord, dvs. en instruktion, et registernavn eller en condition code.

ERROR 02 DOBBELT DEFINERET SYMBOL

Symbolet er defineret tidligere i programmet.

ERROR 03 SYMBOL MANGLER

Symbolet mangler foran et kolon (:) eller begge dele mangler ved en EQU instruktion.

ERROR 04 UKENDT INSTRUKTION

Instruktionen genkendes ikke af assembleren.

ERROR 05 NEDRUNDINGSFEJL

En 8-bitoperands værdi overstiger 255, et bitnummer ved BIT, SET eller RES er større end 7, adressen ved en RST instruktion er ulovlig, eller operanden ved en IM instruktion er ikke 0, 1 eller 2.

ERROR 06 INDEXERINGSFEJL

Det relative offset ved en JR eller RCAL instruktion er mindre end -126 eller større end 129, eller til-lægsværdien ved en indexeringsinstruktion er mindre end -127 eller større end 128, eller til-lægsværdien ved en JP (IX) eller JP (IY) er ikke nul.

ERROR 07 UKENDT SYMBOL

Symbolet der indgår i operanden findes ikke i symboltabellen.

ERROR 08 ULOVLIG OPERAND

Kombinationen af register og symbol eller konstant er ulovlig ved denne instruktion.

ERROR 09 ORG MANGLER

ERROR 10 END MANGLER

ERROR 20 OVERWRITE ERROR

En objektkodebyte genereret af den nuværende linie ville overskrive kildetekstbufferen hvis den blev gemt i lageret. Denne fejl forekommer kun når MEM instruktionen har været i brug.

ERROR 99 MEMORY OVERFLOW

Under editering betyder denne fejlmeddelelse at der ikke er plads i bufferen til at udføre operationen.

Under assemblering betyder ERROR 99 at der ikke er plads til symboltabellen i bufferen.

APPENDIX C -- PSEUDOINSTRUKTIONER

Er instruktioner der kan forstås af assembleren, men ikke af microprocessoren. I kildeteksten benyttes de på samme måde som almindelige instruktioner.

ORG	Sæt startadresse	1 operand
MEM	Gem i lageret	1 operand
EQU	Tilskriv værdi	1 operand
DB	Definer byte(s)	1 eller flere operander
DW	Definer word(s)	1 eller flere operander
DS	Definer lager	1 operand
END	Stop assemblering	Ingen operander

ORG ORIGIN, definerer startadressen på den efterfølgende objektkode, fx.

ORG 1000H

Instruktionen kan benyttes så ofte man vil i programmet.

MEM Lægger objektkoden direkte ud i lageret, startende fra den adresse man specificerer.
Bemærk, at når ORG instruktionen benyttes sættes MEM instruktionen ud af funktion.

EQU EQUAL, tilskriver en værdi til et symbol.

DB * DEFINE BYTE, genererer værdien af operanden eller operanderne ind i objektkoden. Der kan være lige så mange operander man vil, blot de er adskilt af kommaer, fx.

DB OFFH,"B"&0FH,100B,CR

Værdien af hver enkelt operand må ikke overstige 255 (altså een byte).
Ved hjælp af DB instruktionen kan man også generere tekststreng, fx.

TEKST: DB 'Skriv et tal: ',CR

Tekststreng kan have hvilkensomhelst længde, man kan, i modsætning til enkelte bogstaver, ikke indgå i udtryk. Tekststreng skal altid omslutes af enkelte eller dobbelte anførelsestegn.

DW DEFINE WORD, bruges til at generere 16-bits værdier, så som adresser, i objektkoden. Den mindst betydende byte (LSB) bliver lagt i den første byte, og den mest betydende (MSB) i den anden, hvilket er Z80 standard.
Ligesom ved DB instruktionen kan der godt være flere operander, blot de er adskilt af kommaer.

DS DEFINE STORAGE, hopper et antal bytes, svarende til operandens værdi, over i objektkoden.
Denne instruktion benyttes normalt til at definere tekstbuffere etc.

END Bruges til at terminere assembleringen og skal være til stede.

APPENDIX D -- INTERNE REGISTRE

NAP Benytter lageret mellem 0F00H og 1000H som arbejdslager. Følgende adresser i dette område kan være af interesse for brugeren:

0F00H-0F01H	Indeholder kildetekstens startadresse.
0F02H-0F03H	Indeholder kildetekstens slutadresse.
0F04H-0F05H	Indeholder symboltabellens slutadresse.
0F06H-0F07H	Indeholder arbejdsbufferens slutadresse.

APPENDIX E -- BRUGERDEFINERBAR UDSKRIFTSROUTINE

Når man bruger option 'E' ved 'A' eller 'S' kommandoen, kalder NAP den brugerdefinerbare udskriftsrutine i NAS-SYS.

Adressen på denne rutine står i cellerne 0C78H-0C79H (\$UOUT), og den bør følge de regler der er beskrevet i afsnittet INPUT AND OUTPUT i NAS-SYS manualen.

APPENDIX F -- ET PROGRAMEKSEMPEL

```

-----

      ORG  0E00H
      MEM  0E00H

; Dette program giver repeterende
; tastatur

; Monitorsubroutiner

MRET:  EQU  5BH
KBD:   EQU  61H
SRLIN: EQU  70H
NIM:   EQU  72H
UIN:   EQU  76H

; Monitoradresser

KMAP:  EQU  0C01H
UINAD: EQU  0C7BH

; Dette program sætter repetitions-
; programmet i funktion.
; Det kaldes ved at skrive EE00.

START: LD  HL,INKEY      ;Gem addr på user-
      LD  (UINAD),HL    ;subroutinen
      LD  HL,INTBL      ;Sæt den i funktion
      SCAL NIM
      SCAL MRET

KEYS:  DB   0DH,40H      ;Statusord

INTBL: DB   UIN          ;Indlæsningstabel
      DB   SRLIN
      DB   0

SETTBL: DB  10000000B    ;Denne tabel sørger
      DB  10000000B    ;for at ignorere SHIFT
      DB  10000000B    ;CTRL og GRAPH tasterne
      DB  10000000B
      DB  11000000B
      DB  10000000B
      DB  10000000B
      DB  10111000B

; Monitoren kommer her når den
; leder efter en karakter

INKEY: LD  HL,SETTBL     ;Nulstil tastaturets
      LD  DE,KMAP+1     ;statustabel,
      LD  B,8           ;som er 8 bytes lang
RESET: LD  A,(DE)        ;Nulstil en celle
      AND (HL)
      LD  (DE),A
      INC HL            ;Tag næste celle
      INC DE
      DJNZ RESET        ;Gør det 9 gange

```

	SCAL	KBD	;Scan tastaturet
	LD	HL,KEYS	;Peg HL til statusord
	JR	C,KEY	;Hop hvis tast nede
	XOR	A	;ellers nulstil A
NEWKEY:	LD	(HL),A	;Gem nuværende tast
	INC	HL	;Peg HL til tæller
	LD	(HL),40H	;Lang pause
	RET		;Returner
KEY:	CP	(HL)	;Samme som sidst?
	SCF		;Set carryflag
	JR	NZ,NEWKEY	;Nej, returner karakter
	OR	A	;Ja, nulstil carryflag
	INC	HL	;Træk 1 fra tæller
	DEC	(HL)	;Tæller=0?
	RET	NZ	;Nej, returner
	LD	(HL),18H	;Ja, kort pause
	SCF		;Indiker karakter
	RET		;og returner

APPENDIX G -- KOMMANDOOVERSIGT

NAP kommandoer:

A -- Assemblering
 E -- Editering
 H -- Kildetextens adresser
 Q -- Returner til NAS-SYS
 R -- Indlæs fra kassettebåndoptager
 S -- Udlæs symboltabellen
 W -- Udlæs til båndoptager
 Z -- Slet kildeteksten

Editor kommandoer:

CTRL/B	Til første linie i teksten
CTRL/E	Til sidste linie i teksten
CTRL/N	Næste side
CTRL/O	Foregående side
CTRL/F	Find indtastet søgetekst
CTRL/C	Fortsæt søgningen
OP	Op en linie
NP	Ned en linie
SHFT/OP	Slet linie
SHFT/NP	Indsæt linie
ENTER	Ned en linie + Indsæt linie
ESC	Slet linie + Op en linie
VP	En position til venstre
HP	En position til højre
SHFT/VP	Slet karakter
SHFT/HP	Indsæt karakter
BS	Backspace
CH	Fremad tabulering
CS	Baglæns tabulering
LF	Til første position i linien
CTRL/;	Start blok
CTRL/=	Slut blok
CTRL/D	Delete blok
CTRL/I	Insert blok
CTRL/P	Print blok
CTRL/X	Forlad editoren